**Rp-Tumba College**

**Course Machine Learning**

**On 11/March/ 2025**

**QUIZ 2**

**GROUP 6**

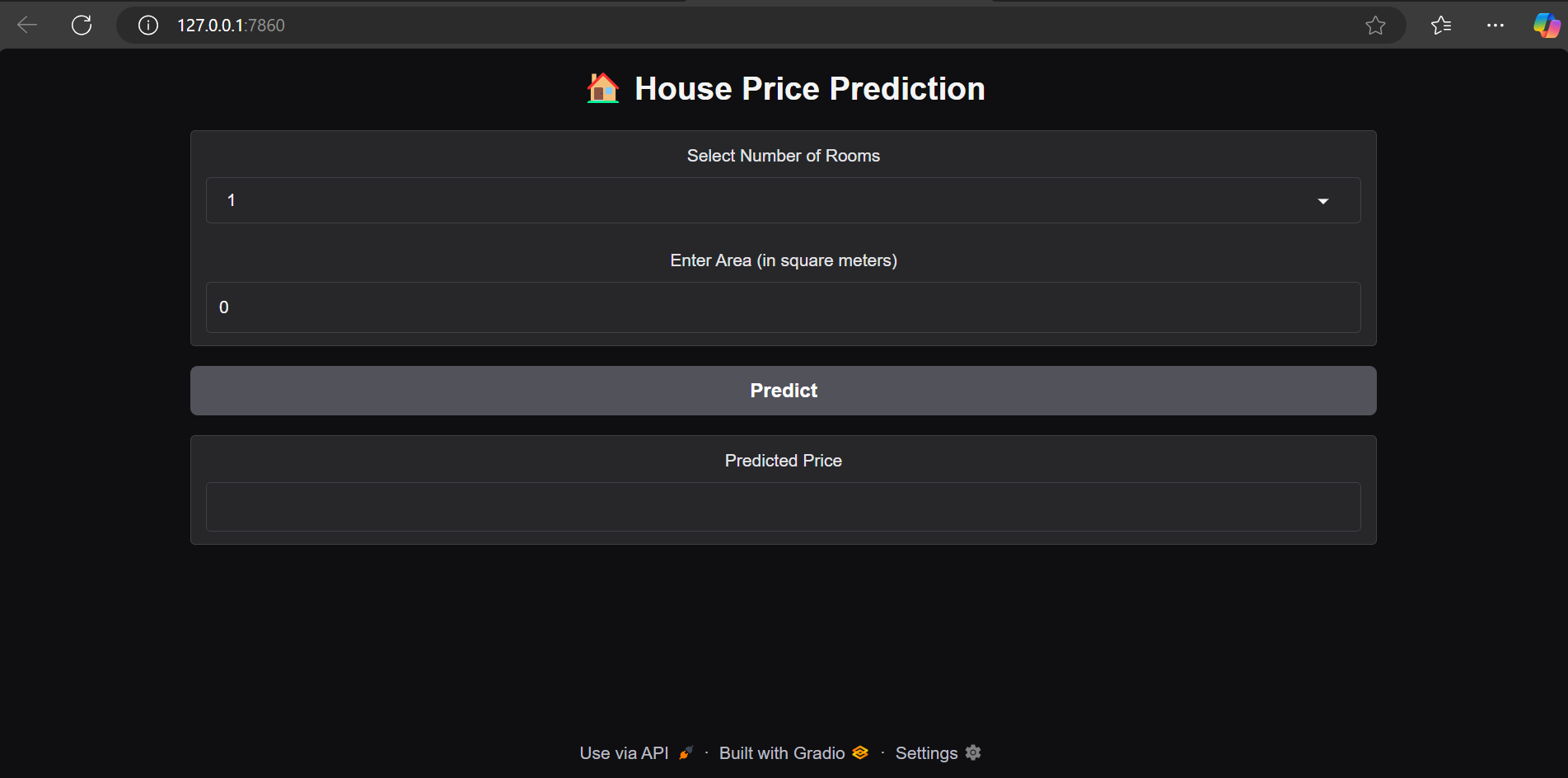
ISHIMWE Phoibe 24RP14903

**House Price Prediction**

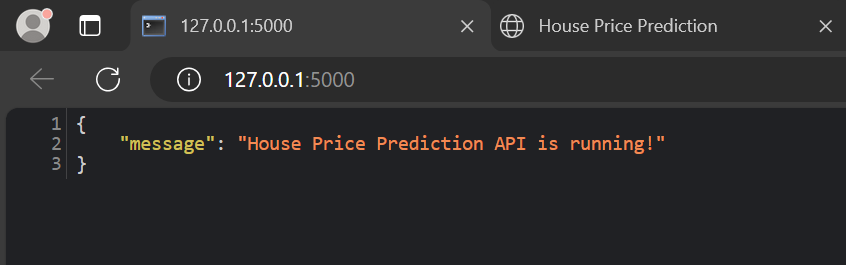
**Overview**

This project is a House Price Prediction system that utilizes a machine learning model to estimate house prices based on the number of rooms and the area size. It consists of a Flask API backend, a Gradio-based frontend, and an additional web-based frontend using HTML and JavaScript.

Gradio\_frontend



**Flask API backend “ backendapi.py”**



**backendapi.py**

**Code:**

from flask import Flask, request, jsonify

from joblib import load

import numpy as np

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

model = load("house\_model.pkl")

@app.route("/")

def home():

    return jsonify({"message": "House Price Prediction API is running!"})

@app.route("/predict", methods=["POST"])

def predict\_price():

    data = request.json

    try:

        num\_rooms = int(data["num\_rooms"])

        area = float(data["area"])

        features = np.array([[area, num\_rooms]])

        prediction = model.predict(features)[0][0] \* 1\_000\_000

        return jsonify({"predicted\_price": f"{prediction:,.0f} Rwf"})

    except Exception as e:

        return jsonify({"error": str(e)}), 400

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**Features**

Flask API: to handle predictions

Gradio interface: for interactive predictions

HTML & JavaScript frontend: for a user-friendly experience

Machine Learning Model: loaded via Joblib for predictions

CORS enabled: for frontend-backend communication

Technologies Used

Backend:

- Flask (API development)

- Flask-CORS (Cross-Origin Resource Sharing)

- Joblib (Model loading)

- NumPy (Data processing)

Frontend:

- Gradio (Interactive UI)

- HTML & JavaScript (Web-based UI)

- CSS (Styling)

**Project Structure**

home.html # Web-based frontend

backendapi.py # Flask API backend

connect.py # Gradio-based frontend fetching from API

index.py # Another Gradio-based interface

house\_model.pkl # Pre-trained ML model

* Installation & Setup
* Prerequisites
* Python 3.x installed
* Flask and required dependencies installed

Install Dependencies

pip install flask flask-cors joblib numpy gradio requests

Running the Flask Backend

python backendapi.py

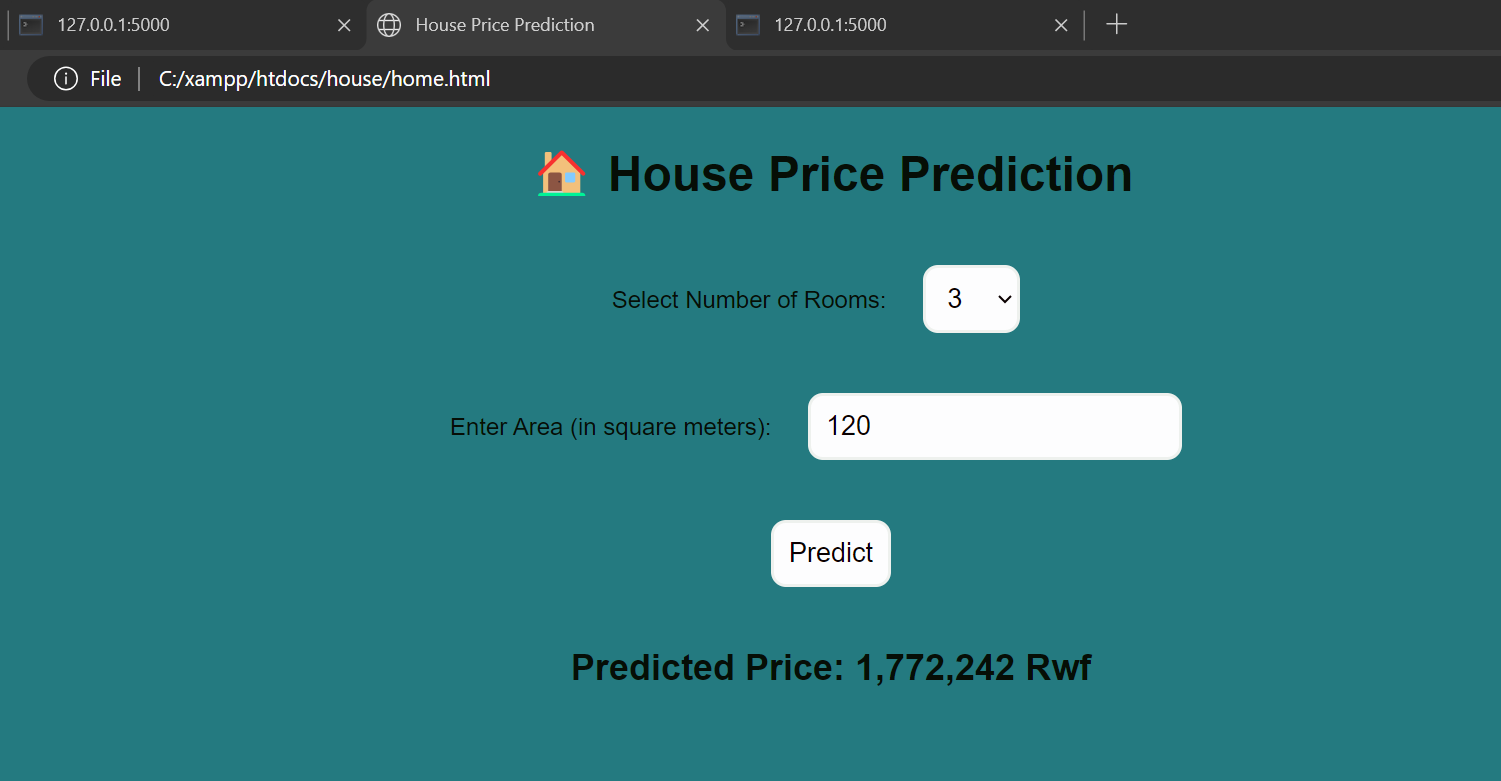
The API will start at `http://127.0.0.1:5000/`.

Running the Web Frontend

- Open “home.html” in a browser

- Enter the number of rooms and area size.

- Click "Predict" to fetch the house price from the API.



Running the Gradio Interface

python backendapi.py

OR

python index.py

API Endpoint

POST: url = "http://127.0.0.1:5000/predict"

Request Body (JSON):

json

{

"num\_rooms": 3,

"area": 120

}

Response (JSON):

json

{

"predicted\_price": "12,000,000 Rwf"

}